Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1.   (Previously presented) A method comprising:

performing, by a software stream, heap memory operations on a first end of a linked list of free heap memory of a heap pile; and concurrently

returning a return block of heap memory, by a hardware device that used the return block of heap memory, to the heap pile at a second end of the linked list of free heap memory.

2.   (Cancelled).

3.   (Previously presented) The method as defined in claim 1 wherein returning a return block of heap memory further comprises:

writing a null to a next block field of the return block of heap memory;

writing a block number of the return block of heap memory to a next block field of a last block of heap memory in the linked list;

changing the contents of a bottom register to point to the return block of heap memory; and thereby

making the return block of heap memory a last entry in the linked list.

4.   (Previously presented) The method as defined in claim 1 wherein performing heap memory operations further comprises returning, by the software stream, a second block of heap memory by placing the second block of heap memory at the first end of the linked list.

Appl. No. 10/037,806
Amdt. dated December 15, 2005
Reply to final Office action of October 7, 2005

5.   (Previously presented) The method as defined in claim 4 wherein returning the second block of heap memory at the first end of the linked list by the software stream further comprises:

determining a block number of a primary block of heap memory resident at the first end of the linked list;

writing the block number of the primary block of heap memory to a next block field of the second block; and

writing atomically a block number of the second block to a top register.

6.   (Previously presented) The method as defined in claim 5 wherein determining a block number of a primary block of heap memory resident at the first end of the linked list further comprises reading the top register prior to writing the block number of the second block.

7.   (Previously presented) The method as defined in claim 1 wherein performing heap memory operations further comprises removing, by the software stream, heap memory from the first end of the linked list.

8.   (Previously presented) The method as defined in claim 7 wherein removing heap memory from the linked list heap management system further comprises taking a primary block of heap memory resident at the first end of the of the linked list.

9.   (Previously presented) The method as defined in claim 8 wherein taking a primary block of heap memory further comprises:

determining a block number of the primary block;

reading a next block field of the primary block of memory; and

removing the primary block if the next block field of the primary block does not indicate a null.

10.   (Original) The method as defined in claim 9 wherein determining a block number of the primary block further comprises reading a top register, wherein the top register identifies the beginning of the linked list.

11.   (Original) The method as defined in claim 9 wherein removing the primary block if the next block field of the primary block does not indicate a null further comprises writing a block number of the next block field of the primary block to the top register.

12.   (Currently amended) A method of managing a heap memory comprising:
    maintaining unused blocks of heap memory as a linked list, and wherein
        the unused blocks of the linked list comprise a first block at a
        beginning of the linked list, a second block pointed to the first block,
        and a third block at an end of the linked list;
    removing, by a software stream using an atomic operation, the first block
        from the linked list, thus making the second block the beginning of
        the linked list; and
    returning a return block, by a hardware device that used the return block,
        to the linked list by placing the return block at the end of the linked
        list with a non-atomic operation.

13.   (Previously presented) The method of managing a heap memory as defined in claim 12 wherein returning a return block further comprises:
    writing a null to a next block field of the return block;
    reading a bottom register, the bottom register identifying the third block;
    writing a block number of the return block to a next state field of the third
        block; and
    writing the block number of the return block to the bottom register.

14.   (Original) The method of managing a heap memory as defined in claim 12 wherein removing, by a software stream, the first block from the linked list further comprises:

> reading a top register, the top register identifying the first block;
>
> reading a next block field of the first block, the next block field of the first block identifying the second block; and
>
> writing a block number of the second block to the top register.

15.   (Original) The method of managing a heap memory as defined in claim 14 wherein writing a block number of the second block to the top register further comprises atomically writing the block number of the second block to the top register.

16.   (Original) The method of managing a heap memory as defined in claim 12 further comprising returning, by a software stream, a fourth block to the linked list by placing the fourth block at the beginning of the linked list, thus making the fourth block the beginning of the linked list.

17.   (Original) The method of managing a heap of memory as defined in claim 16 wherein returning a fourth block to the linked list by placing the fourth block at the beginning of the linked list further comprises:

> reading a top register, the top register identifying the beginning of the linked list;
>
> writing a block number of the block identified by the top register to a next state field of the fourth block; and
>
> writing a block number of the fourth block to the top register.

18.   (Original) The method of managing a heap memory as defined in claim 17 wherein writing a block number of the fourth block to the top register further comprises atomically writing the block number of the fourth block to the top register.

19.   (Currently amended) A method of managing a heap memory in a computer system, the method comprising:

        allowing a software thread <u>executed on a processor</u> to add and remove blocks of heap memory from a linked list of free blocks of heap memory in a last-in/first-out (LIFO) fashion at a first end of the linked list; and

        allowing a hardware device that uses blocks of heap memory to add the blocks of heap memory to the linked list of free blocks of heap memory at a second end of the linked list, <u>the hardware device coupled to the processor by way of a communication bus</u>.

20.   (Original) The method of managing a heap memory in a computer system as defined in claim 19 wherein allowing a software thread to remove blocks of heap memory in LIFO fashion further comprises:

        determining, by the software thread, a block number of a block of heap memory at the first end of the linked list; and

        removing the block of heap memory at the first end of the linked list.

21.   (Original) The method of managing a heap memory in a computer system as defined in claim 20 determining a block number of a block of heap memory at the first end of the linked list further comprises reading a beginning register that stores a block number of a block of heap memory at the first end of the linked list.

22.   (Original) The method of managing a heap memory in a computer system as defined in claim 21 wherein removing the block of heap memory at the first end of the linked list further comprises:

        reading a next block field of the block of heap memory at the first end of the linked list to identify a block number of a next block in the linked list; and

        writing the block number of the next block in the linked list to the beginning register.

23.   (Original) The method of managing a heap memory in a computer system as defined in claim 20 wherein allowing a software thread to add blocks of heap memory in LIFO fashion further comprises:

  determining, by the software thread, a block number of a block of heap memory at the first end of the linked list;

  writing the block number of the block of heap memory at the first end of the linked list to a next block field of a return block of heap memory; and

  making the return block of heap memory the first end of the linked list.

24.   (Original) The method of managing a heap memory in a computer system as defined in claim 23 wherein determining a block number of a block of heap memory at first end of the linked list further comprises reading a beginning register that stores a block number of a block of heap memory at the first end of the linked list.

25.   (Original) The method of managing a heap memory in a computer system as defined in claim 24 wherein making the return block of heap memory the first end of the linked list further comprises writing a block number the return block of heap memory to the beginning register.

26.   (Previously presented) The method of managing a heap memory in a computer system as defined in claim 20 wherein allowing a hardware device that uses blocks of heap memory to add the blocks of heap memory to the linked list of free blocks of heap memory at a second end of the linked list further comprises:

  determining, by the hardware device, a block number of a block of heap memory at the second end of the linked list;

  writing, by the hardware device, a block number of a return block of heap memory to a next block field of the block of heap memory at the second end of the linked list; and

making the return block of heap memory the second end of the linked list.

27.   (Original) The method of managing a heap memory in a computer system
as defined in claim 26 wherein determining a block number of a block of heap
memory at the second end of the linked list further comprises reading an end
register that stores a block number of the block of heap memory at the second
end of the linked list.

28.   (Original) The method of managing a heap memory in a computer system
as defined in claim 27 wherein making the return block of heap memory the
second end of the linked list further comprises writing a block number the return
block of heap memory to the end register.

29.   (Previously presented) A computer system comprising:
    a microprocessor executing a software stream;
    a main memory array, a portion of the main memory array allocated to be
        a heap memory, and wherein unused portions of the heap memory
        are part of a heap pile, the heap pile further comprising
            a plurality of blocks;
            each block having a next block field; and
            wherein the heap pile is maintained as a linked list, each
                block's next block field pointing to a next block in the
                list;
    a first bridge logic device coupling the microprocessor to the main memory
        array;
    a hardware device coupled to the heap memory through the first bridge
        logic device;
    wherein the software stream executed on the microprocessor removes
        blocks of heap memory from a beginning of the heap pile; and
        simultaneously

the hardware device returns blocks of heap memory used by the hardware device to an end of the heap pile.

30.   (Previously presented) The computer system as defined in claim 29 wherein the plurality of blocks each have the same number of bytes.

31.   (Cancelled).

32.   (Previously presented) The computer system as defined in claim 29 further comprising the software stream returns blocks to the heap pile at the beginning of the heap pile simultaneously as the hardware device returns blocks of heap memory used by the hardware device to the end of the heap pile.

33.   (Currently amended) The computer system as defined in claim 29 wherein the hardware device is one or more selected from the group: the-a graphics card; a network interface card; an audio card; a mass storage device; a hard drive; or a compact disk storage device.

34.–38. (Cancelled).

39.   (Currently amended) A method comprising:
        performing, by a software stream executed on a processor, heap memory
            operations on a first end of a linked list of free heap memory of a
            heap pile; and concurrently
        returning a return block of heap memory to the heap pile at a second end
            of the linked list of free heap memory, the returning by a hardware
            device coupled to the processor selected from group consisting of a
            graphics card, a network interface card, an audio device or a mass
            storage device.

40.   (New) The method comprising as defined in claim 1 wherein performing further comprises performing by a software stream executed on a processor different than the hardware device.

41.   (New) The method as defined in claim 1 further comprising:
        wherein performing further comprises performing by a software stream executed on a processor; and
        wherein returning further comprises returning by a hardware device coupled to the processor by way of a system bus.

42   (New) The method as defined in claim 1 further comprising:
        wherein performing further comprises performing by the software stream using atomic operations on the first end of the linked list; and
        wherein returning further comprises returning by the hardware device using non-atomic operations.